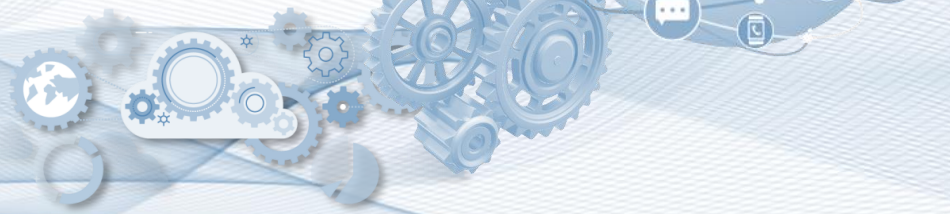
Τμήμα Ηλεκτρολόγων Μηχανικών
& Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΤΕΧΝΙΚΕΣ ΠΡΟΓΡΑΜΜΑΤΙΜΟΥ ΥΠΟΛΟΓΙΣΤΩΝ

Διάλεξη ΙΙ

Το Τελικό Βήμα με Scratch – Μεταβλητές & Πρότυπα

Γιάννης Τζήμας, Καθηγητής



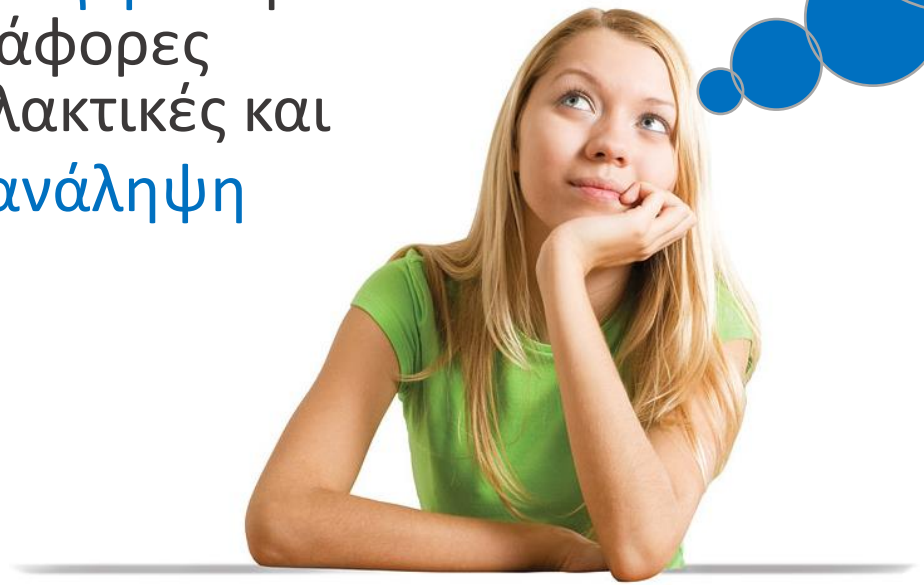
Τι θα δούμε σήμερα…

- Προχωρώντας θα δούμε πιο αναλυτικά τις έννοιες των:
  - μεταβλητών
  - και των προτύπων.

Τι είδαμε στα προηγούμενα μαθήματα και τι μας λείπει;

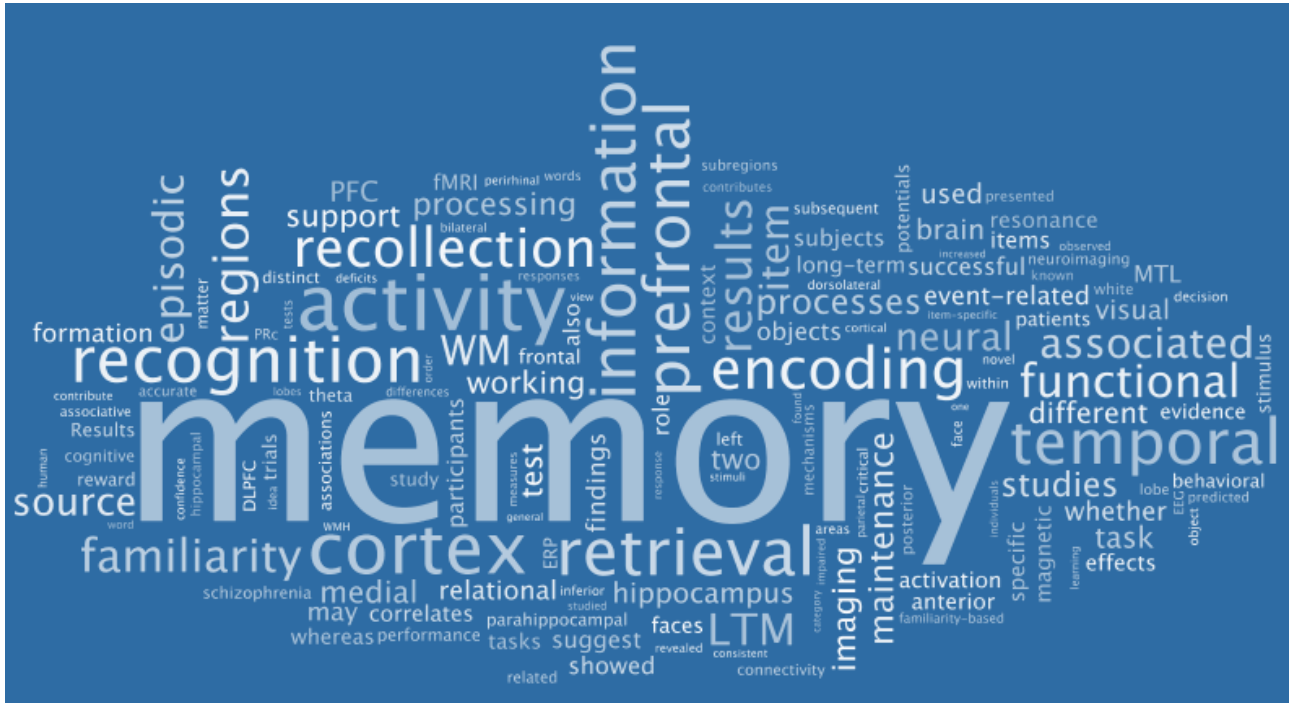
- Οι τρεις προγραμματιστικές δεξιότητες που θα πρέπει να καλλιεργήσετε όπως έχουμε πει είναι:
  - Η αλληλουχία των εντολών
  - Η επιλογή ανάμεσα σε διάφορες εναλλακτικές και
  - Η επανάληψη

Ποιο είναι το **επόμενο βήμα;**
Στην πράξη χρειαζόμαστε επιπλέον:
- Ένα τρόπο για να **θυμόμαστε πράγματα** – "**variables – μεταβλητές**"
- Δεξιότητες για **επίλυση προβλημάτων** – "**patterns - πρότυπα**"



Μνήμη - Memory

- Όπως πιθανά ήδη ξέρετε από άλλα σας μαθήματα, οι ηλεκτρονικοί υπολογιστές (PCs, laptops, mobiles κ.λπ.) έχουν αρκετή μνήμη για την επεξεργασία δεδομένων.
  - 4GB,
  - 8GB,
  - 32GB,
  - Ίσως και περισσότερο!



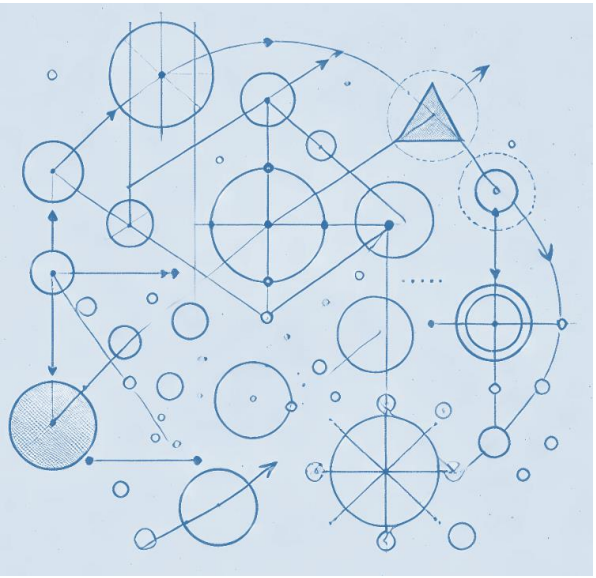
Τύποι Μνήμης

- Μόνιμη
  - Τα δεδομένα υπάρχουν ακόμα και αν η συσκευή κλείσει.
  - π.χ. σκληροί δίσκοι, memory cards κλπ.

- Προσωρινή
  - Τα δεδομένα αποθηκεύονται προσωρινά ώστε να γίνει η απαραίτητη επεξεργασία.
  - π.χ. RAM – Random Access Memory

Μεταβλητές (Variables) = Μνήμη (Memory)

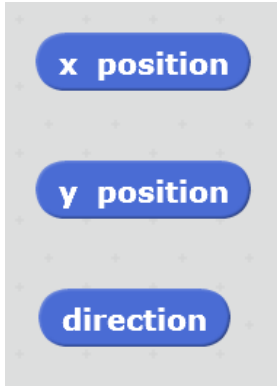
- Οι μεταβλητές είναι η **μνήμη** του **προγράμματός σας**.
  - Τις χρησιμοποιούμε για να αποθηκεύσουμε πληροφορίες όσο το πρόγραμμα «τρέχει».
- **Γιατί** χρειαζόμαστε μεταβλητές;
  - Μερικά παραδείγματα…
    - Για να κάνουμε μαθηματικούς υπολογισμούς – «η απόσταση έως την άκρη της σκηνής είναι…»
    - Για να παρακολουθούμε διάφορα πράγματα – «το σκορ του παιχνιδιού είναι…»
    - Για να αποθηκεύουμε τιμές εισόδου από το χρήστη – «Παρακαλώ πληκτρολογήστε το όνομά σας…»
    - Για να κρατάμε τιμές εξόδου που μπορεί να δείξουμε στην οθόνη – «Η σωστή απάντηση στην ερώτηση είναι…»

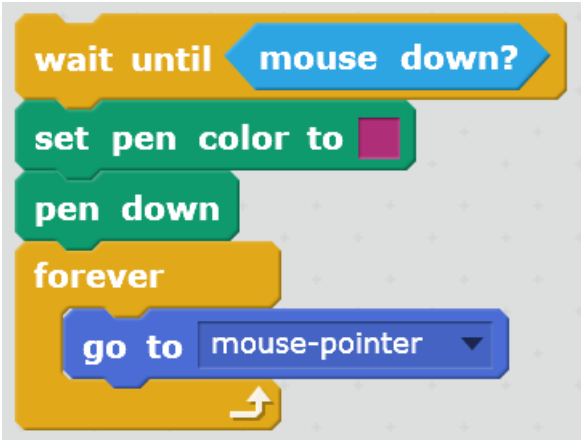


- Ενδιαφέρουσα σημείωση:
  - Στις περισσότερες γλώσσες προγραμματισμού (π.χ. Java, C++, C# κλπ.), οι μεταβλητές χρησιμοποιούν την προσωρινή μνήμη – οι πληροφορίες χάνονται με τον τερματισμό του προγράμματος.
  - Στο Scratch οι μεταβλητές χρησιμοποιούν τη μόνιμη.

Παραδείγματα στο Scratch

- Έχουμε ήδη χρησιμοποιήσει **προκαθορισμένες** μεταβλητές στα προγράμματα που φτιάξαμε…
  - x position
  - y position
  - direction



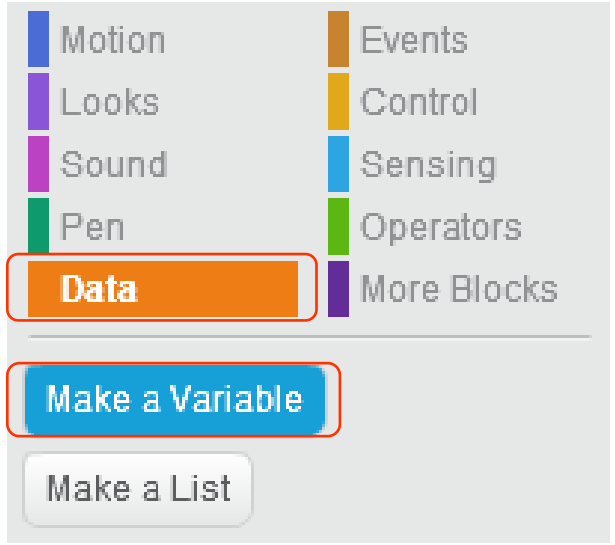


  - pen color
  - pen state (up? down?)
  - mouse position
  - mouse state
  - volume
  - ….

Δημιουργώντας τις Μεταβλητές σας…

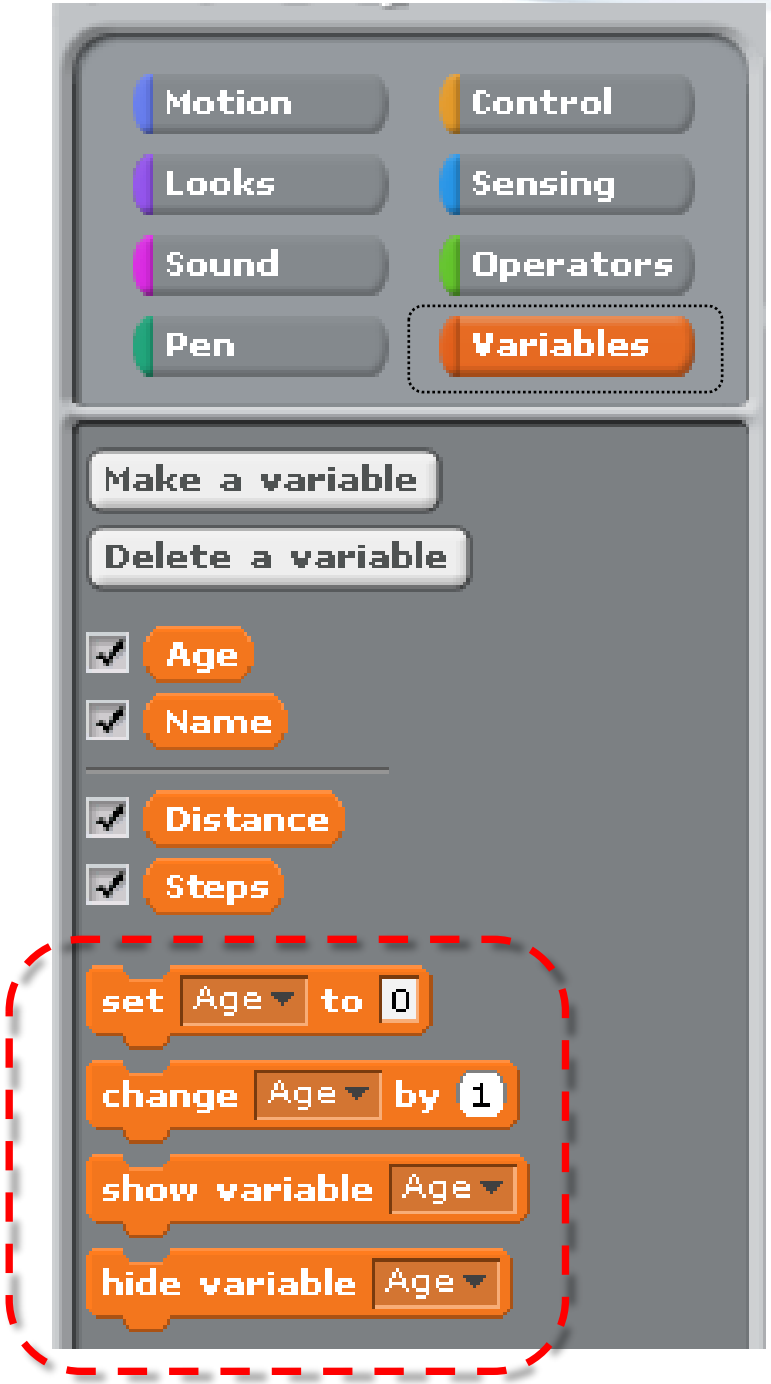
- Δημιουργείστε μία μεταβλητή όταν θέλετε να παρακολουθήσετε κάτι…

- Οι μεταβλητές έχουν διάφορα χαρακτηριστικά:
  - Κάθε μεταβλητή έχει **όνομα**.
  - Μία μεταβλητή περιέχει μία **τιμή**.
  - Το **περιεχόμενο** μίας μεταβλητής μπορεί να **αλλάξει**.
  - Μία μεταβλητή συσχετίζεται με ένα (τοπική/**local** variable), ή με όλα τα sprites (καθολική/**global** variable).
  - Μία μεταβλητή μπορεί να φαίνεται (π.χ. το σκορ), ή να μη φαίνεται στη σκηνή.

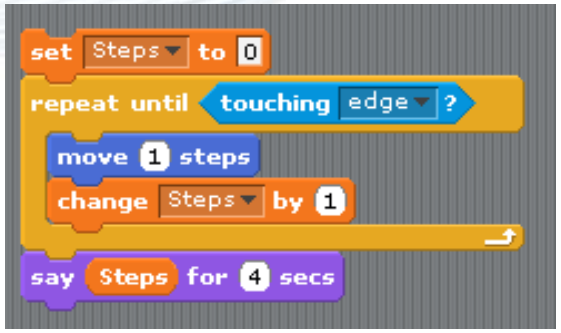


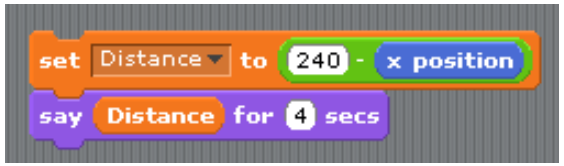
Ένα μικρό demo…

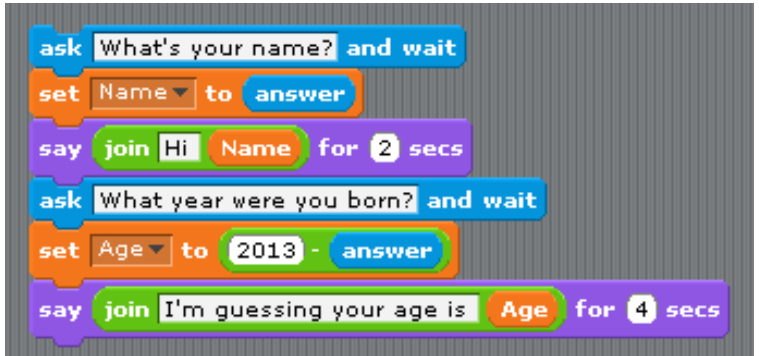
- Δημιουργώντας μεταβλητές:
  - Πόσα βήματα πρέπει να γίνουν έως η Scratch να αγγίξει τη γωνία;
  - Ποια είναι η απόσταση από τη δεξιά πλευρά;
  - Ποιο είναι το όνομα του χρήστη (επώνυμο, όνομα);
  - Ποιο είναι το όνομα και η χρονιά που γεννήθηκε ο χρήστης;
    - Υπολογισμός της ηλικίας

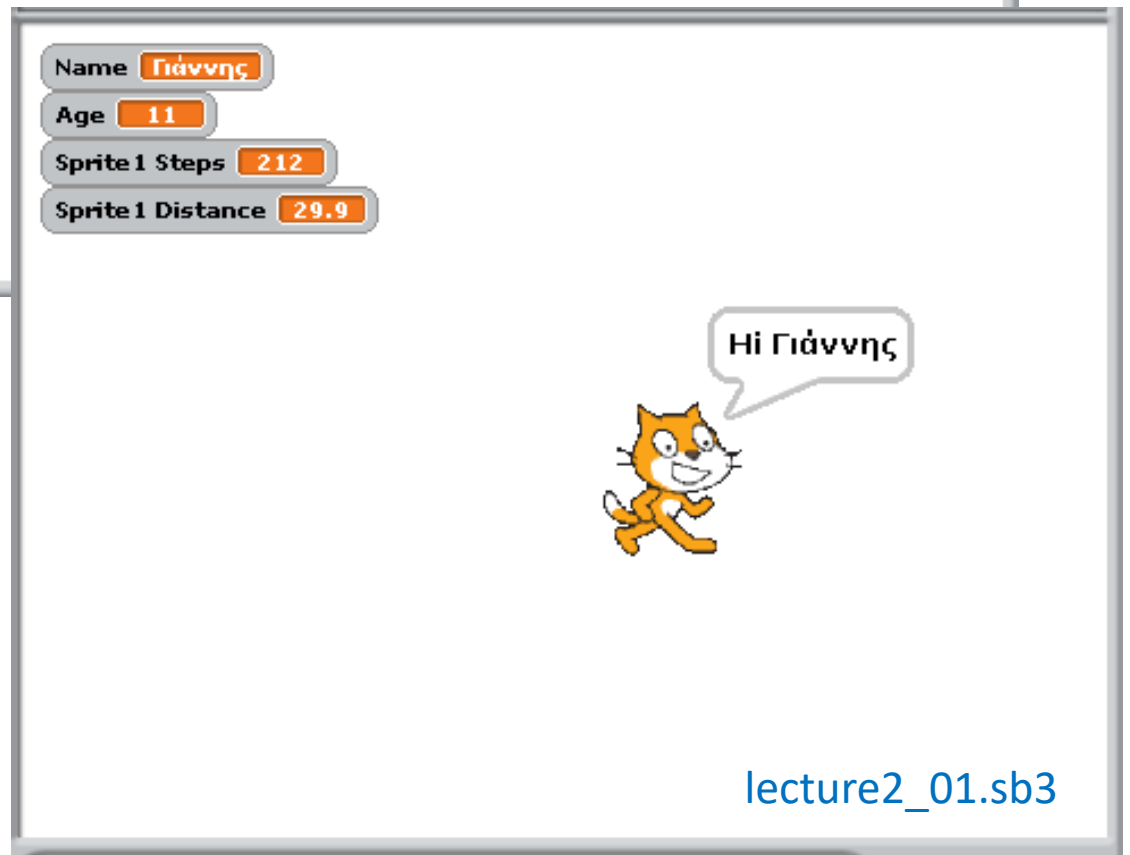


Στην πράξη…





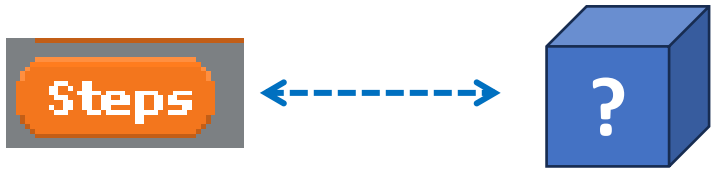




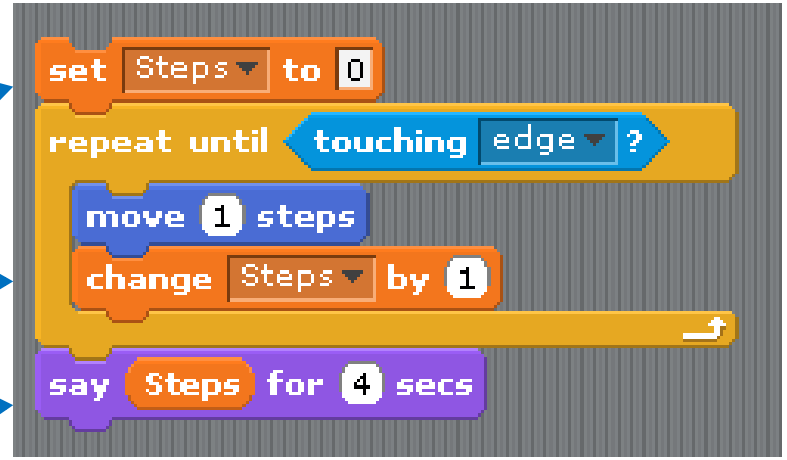
lecture2_01.sb3

Όταν χρησιμοποιείτε λοιπόν μεταβλητές…

- Να σκέφτεστε τη μεταβλητή σαν ένα **κουτί χωρίς αρχική τιμή**:



  - Θα πρέπει να δώσετε μία **αρχική τιμή**,
  - καθώς το πρόγραμμα τρέχει να ενημερώνετε την τιμή αυτή
  - και στο τέλος να τη χρησιμοποιείτε, ή να την παρουσιάζετε.



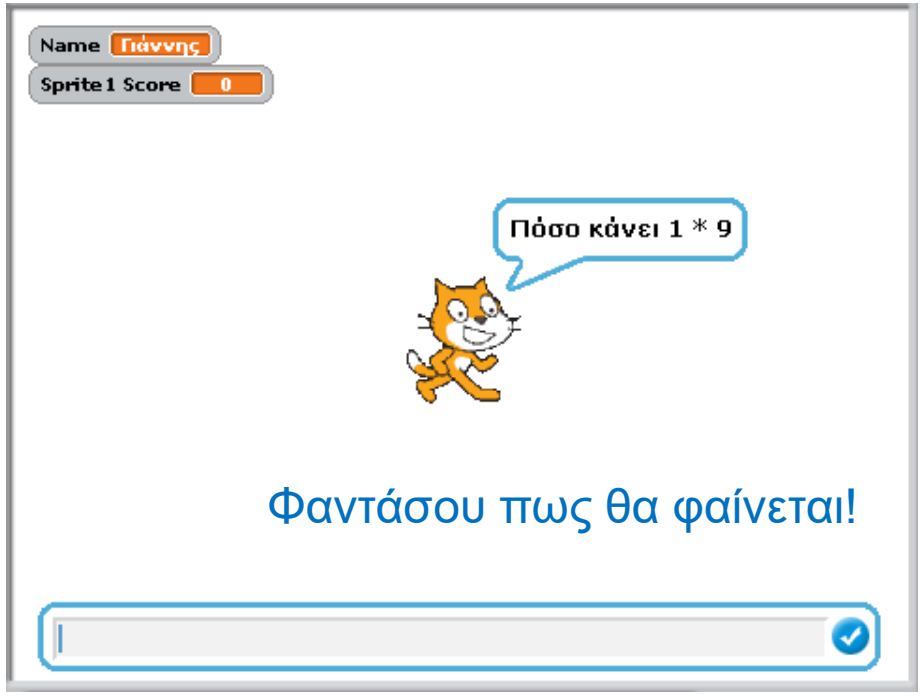
Χρυσός Κανόνας: Χρησιμοποιείτε μία μεταβλητή για κάθε τμήμα δεδομένων που θα πρέπει να παρακολουθείτε!

Ένα ολοκληρωμένο παράδειγμα…

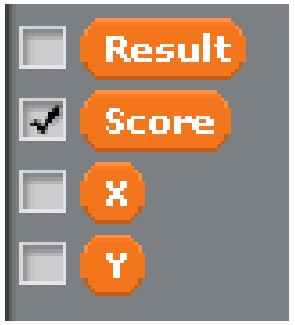
- Ας χρησιμοποιήσουμε όλα όσα μάθαμε μέχρι στιγμής για να φτιάξουμε ένα απλό παιχνίδι πολλαπλασιασμού:
- Το πρόγραμμα θα κάνει τα ακόλουθα:
  - Επιλέγει δύο τυχαίους αριθμούς Χ και Υ
  - Υπολογίζει το σωστό αποτέλεσμα “= Χ * Υ”
  - Στη συνέχεια ρωτάει τον παίκτη: “Πόσο κάνει Χ * Υ;”
  - Συγκρίνει την απάντηση με το σωστό αποτέλεσμα
  - Αν είναι σωστή η απάντηση, τότε ο παίκτης παίρνει ένα βαθμό
  - Επανάληψη…
  - Όταν το παιχνίδι τελειώνει εμφανίζεται το σκορ

Σχεδιάζοντας τη λύση…

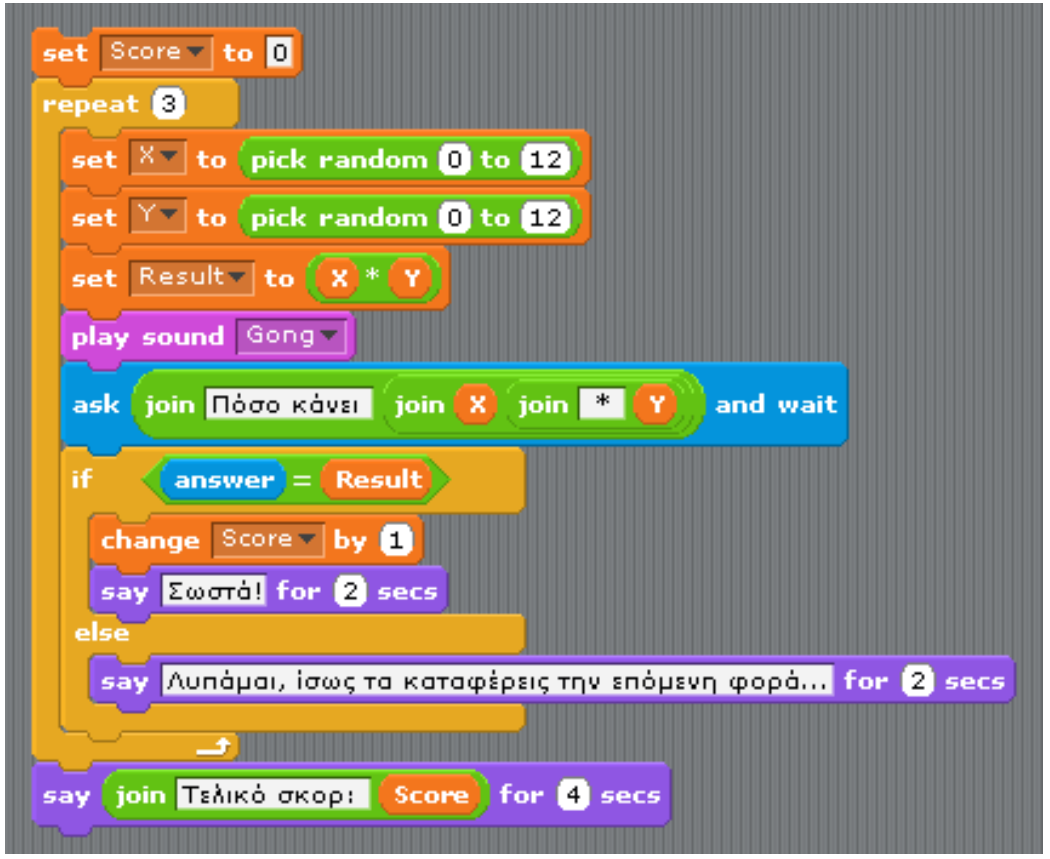
- Επανέλαβε 3 φορές:
  - Επέλεξε δύο τυχαίους αριθμούς, και ονόμασέ τους X και Y
  - Υπολόγισε το σωστό αποτέλεσμα “= X * Y”
  - Ρώτα τον παίκτη: “Πόσο κάνει X * Y;”
  - Σύγκρινε την απάντηση με το σωστό αποτέλεσμα
  - Αν είναι σωστή η απάντηση, τότε ο παίκτης παίρνει ένα βαθμό
- Όταν το παιχνίδι τελειώνει εμφανίζεται το σκορ



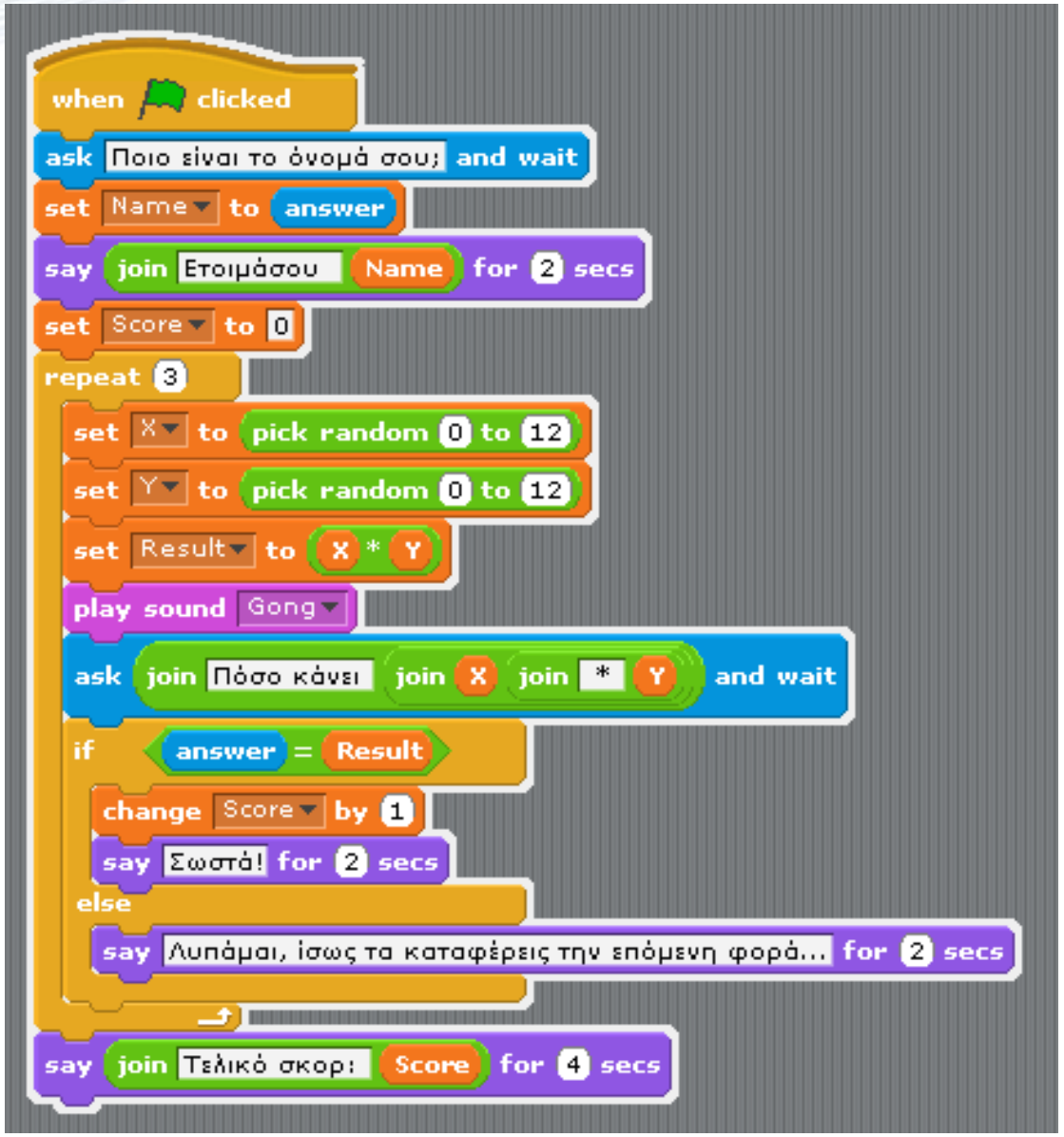
- Πόσες μεταβλητές χρειαζόμαστε;
  - Τουλάχιστον τέσσερις
  - X, Y, Σωστό Αποτέλεσμα, Σκορ



- Η Αλληλουχία:



Συνολικά



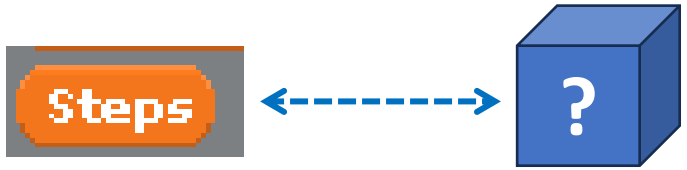
lecture3_demo2.sb3

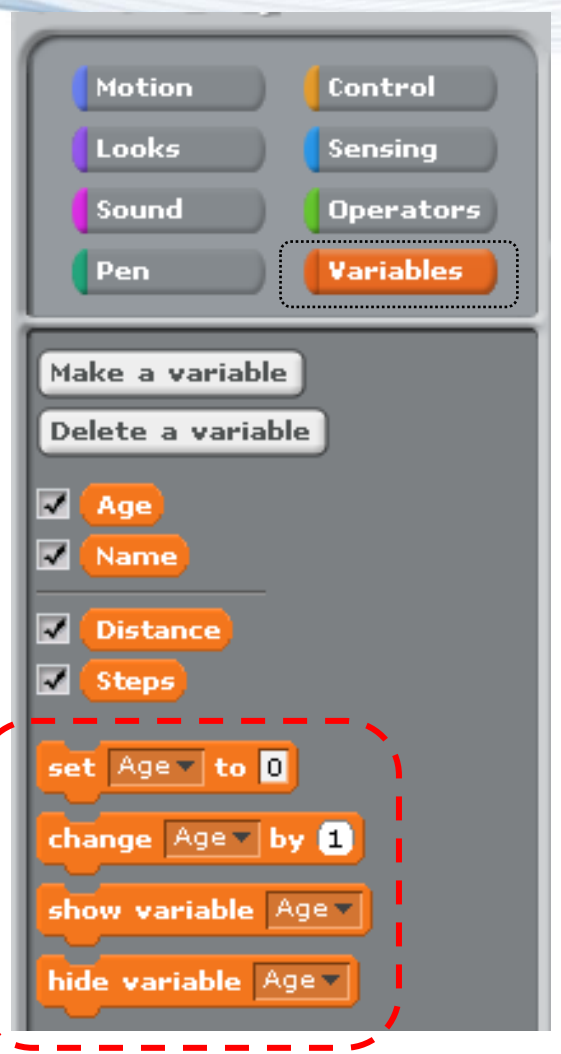
Τι είδαμε μέχρι τώρα;

- Οι μεταβλητές είναι η μνήμη του προγράμματος:
  - Αποθηκεύουν πληροφορίες ενώ το πρόγραμμα «τρέχει».
  - Στο Scratch, οι μεταβλητές δε χρησιμοποιούν την προσωρινή μνήμη, σε αντίθεση με άλλες γλώσσες.

- Σημαντικός κανόνας:
  - Χρησιμοποιείτε μία μεταβλητή για κάθε τμήμα δεδομένων που θα πρέπει να παρακολουθείτε!

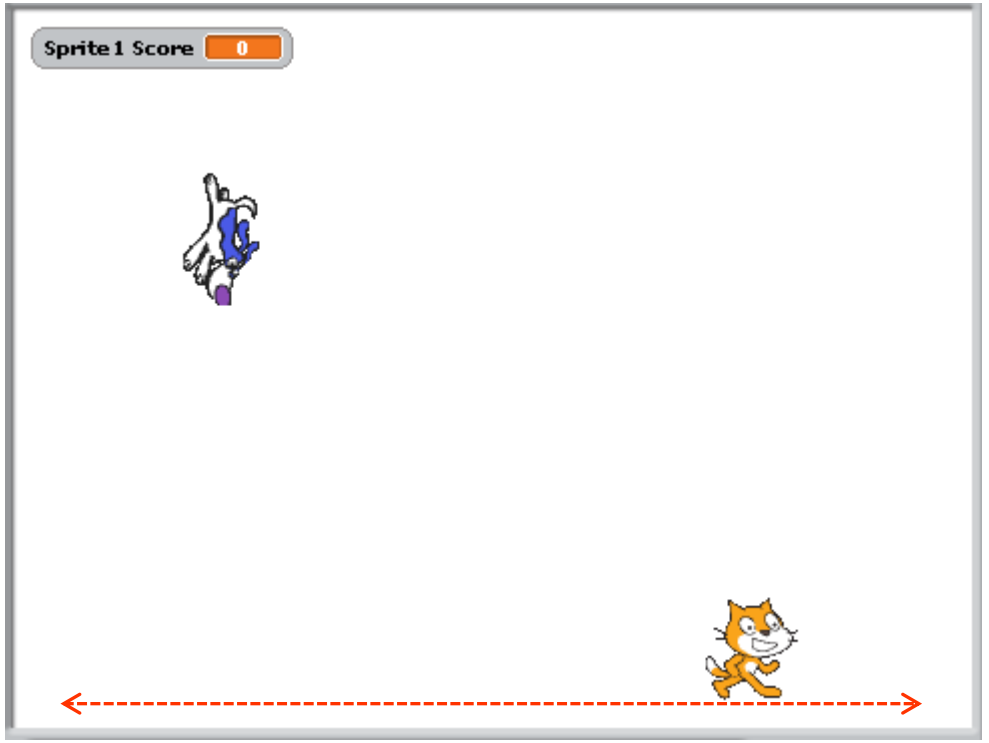
- **Μην ξεχνάτε**:
  - **Αρχικοποιείτε** τις μεταβλητές πριν την πρώτη χρήση τους…



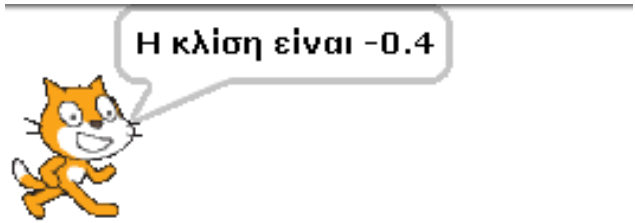


Και λίγη δουλειά για το σπίτι... (1/2)

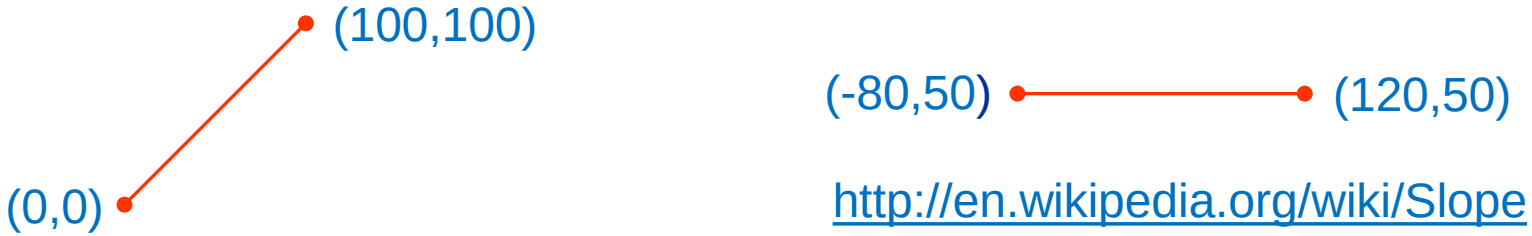
- Άσκηση 1:
  - Η πρώτη βασίζεται στην εργασία του προηγούμενου μαθήματος που φτιάξαμε ένα πρόγραμμα που έβρεχε γάτες και σκύλους.
  - Αυτή τη φορά θέλουμε η Scratch να μπορεί να μετακινηθεί αριστερά και δεξιά στο κάτω μέρος της σκηνής, προσπαθώντας να πιάσει ένα σκυλάκι το οποίο πέφτει από πάνω προς τα κάτω. Για κάθε σκυλάκι που πιάνει η Scratch θα κερδίζει ένα πόντο.



Και λίγη δουλειά για το σπίτι... (2/2)



- Άσκηση 2:
  - Υπολογίστε την κλίση μίας γραμμής. Η κλίση συντίθεται από δύο πράγματα:
    - Πόσο απότομη είναι και
    - Από το αν ανεβαίνει ή κατεβαίνει
  - Για παράδειγμα η γραμμή αριστερά έχει κλίση 1, ενώ η γραμμή δεξιά 0.



http://en.wikipedia.org/wiki/Slope

- Δεδομένων δύο σημείων (x1, y1) και (x2, y2) η κλίση υπολογίζεται από τον ακόλουθο τύπο:
  slope=(y2-y1)/(x2-x1)
- Γράψτε ένα πρόγραμμα που δέχεται σαν είσοδο τα x1, y1, x2 και y2 ζωγραφίζει μία γραμμή και υπολογίζει την κλίση της.
  Προσοχή στις κάθετες γραμμές!

Προγραμματιστικές Δεξιότητες

- Σε λογικό επίπεδο:
  - Αλληλουχία των εντολών
  - Επιλογή ανάμεσα σε διάφορες εναλλακτικές
  - Επανάληψη
- Σε πρακτικό επίπεδο:
  - Μεταβλητές
  - Εμπειρία
  - Αποφασιστικότητα
    - Οι Η/Υ μπορεί να σας δημιουργήσουν απογοήτευση, αλλά…

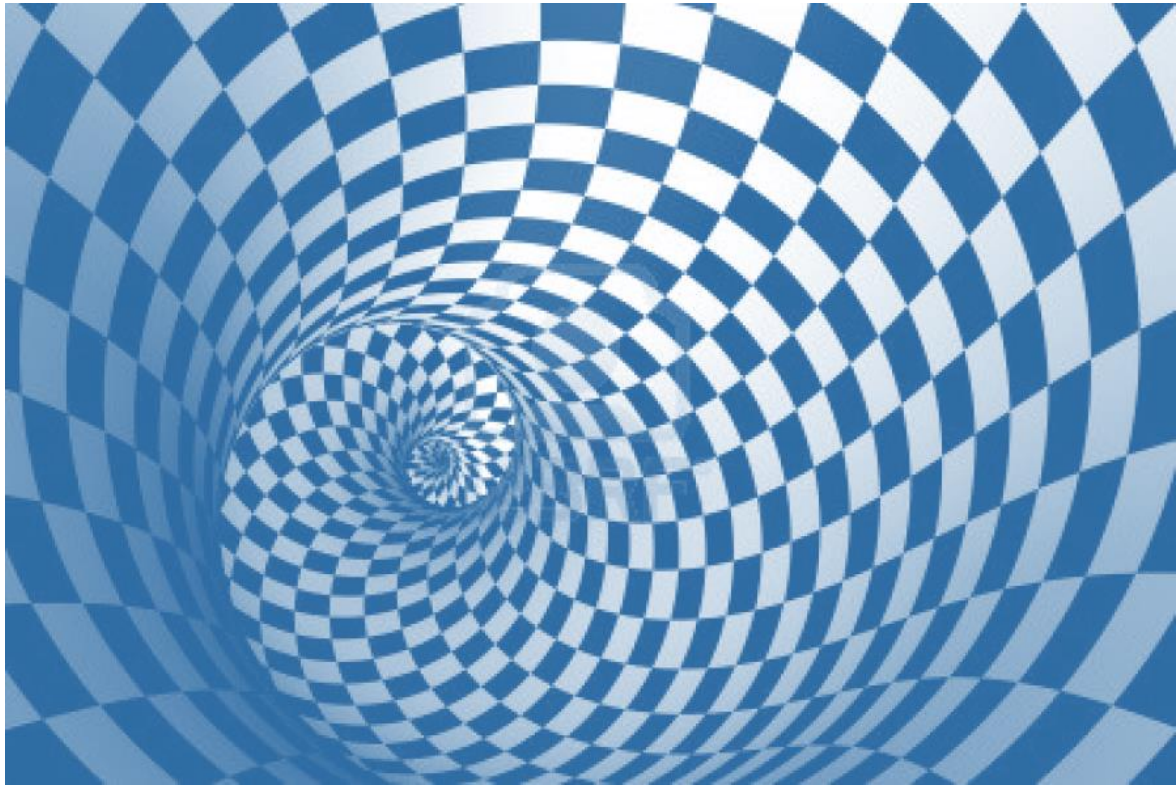
      μην τα παρατήσετε!

Πρότυπα (patterns) = Εμπειρία (experience)

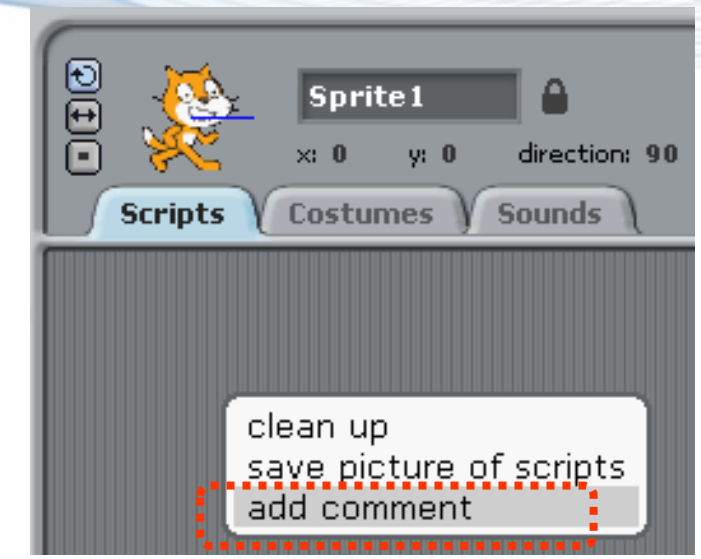
- Τα πρότυπα είναι ένας τρόπος σύλληψης της συλλογικής εμπειρίας άλλων.
  - "**standing on the shoulders of giants**"

- Απλά patterns:
  - commenting on your programs
  - centralized code
  - regression testing

- Μερικά πιο περίπλοκα:
  - game play
  - count, sum, and compute

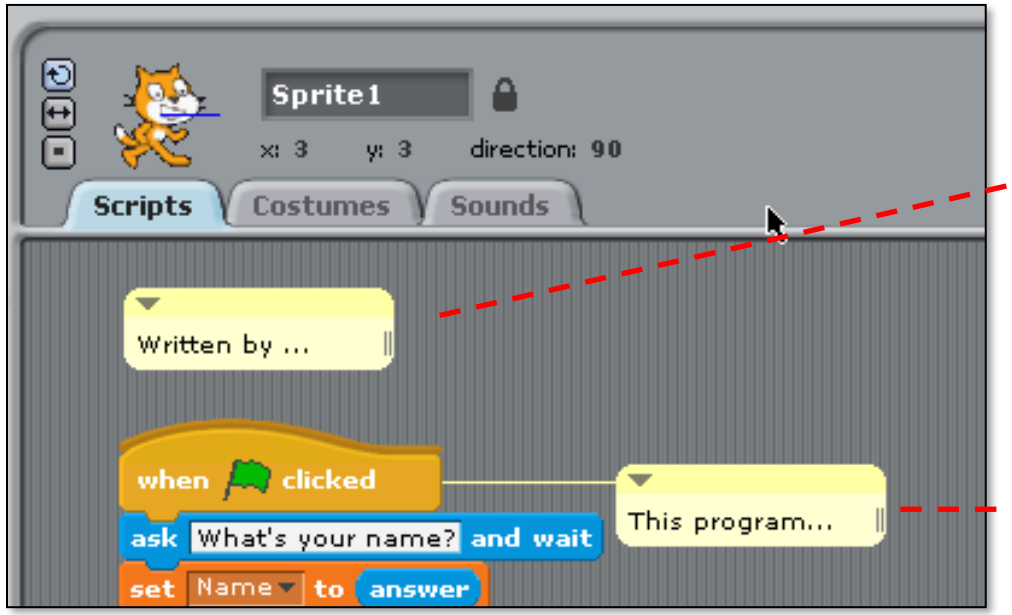


#1 Σχολιασμός



- Καθώς τα προγράμματα γίνονται όλο και πιο περίπλοκα είναι πολύ δύσκολο να θυμάστε όλες τις λεπτομέρειες!
  - Μην το προσπαθήστε καν!

- Αντί για αυτό, προσθέστε **σχόλια**
  - Τα σχόλια είναι ένας τρόπος μόνιμης καταγραφής των σκέψεών σας, ως μέρος του προγράμματος
  - Τα σχόλια αγνοούνται από το Scratch και από όλες τις γλώσσες προγραμματισμούς – είναι φτιαγμένα για να τα διαβάζουμε εμείς.

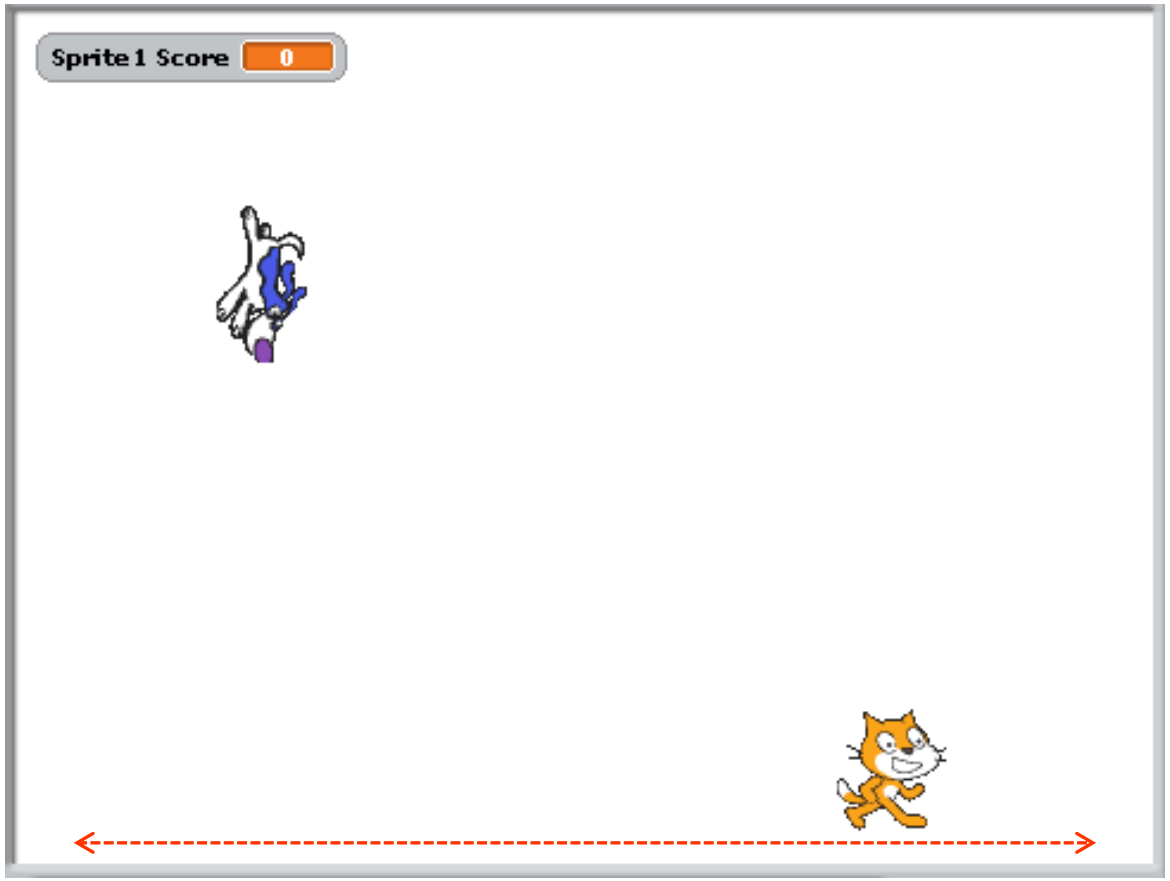


standalone

associated with a block

Ένα μικρό demo…

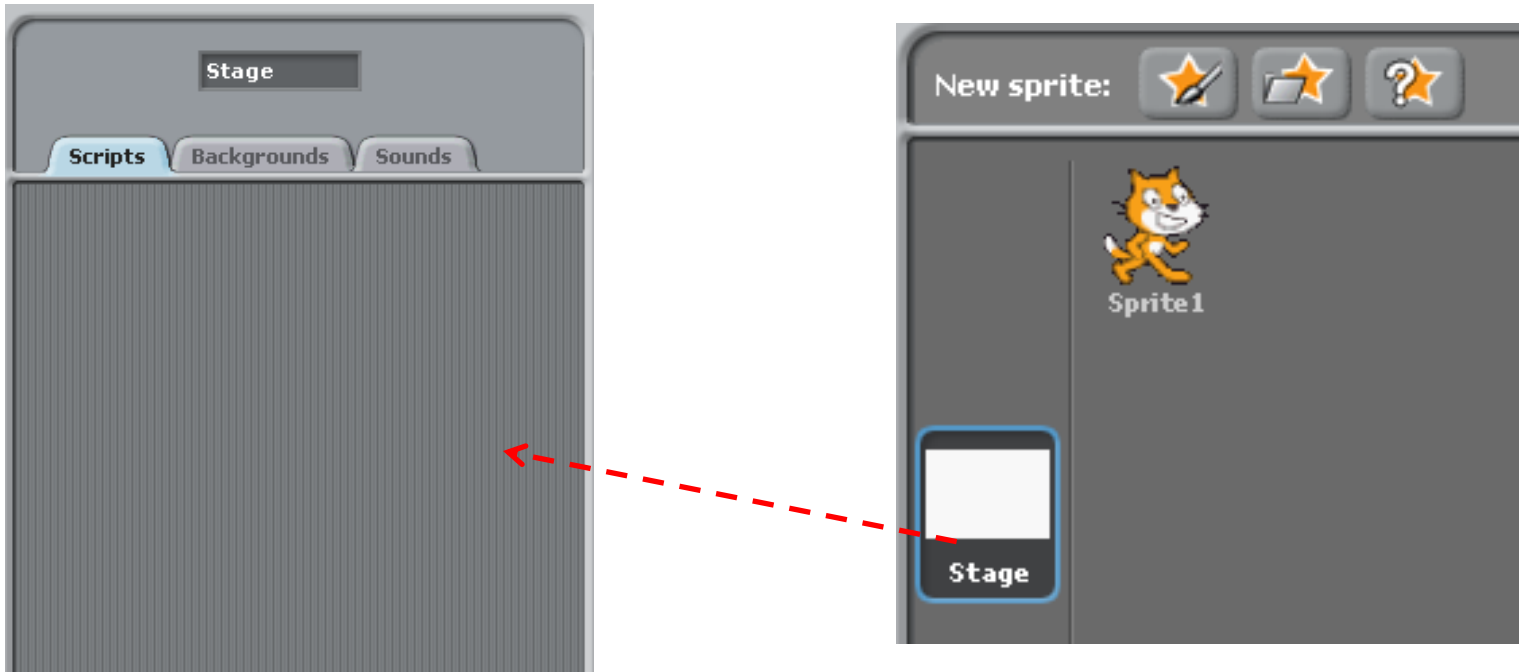
- Από την πρώτη άσκηση…



#2 Centralized Code

- Κάποιος κώδικάς μας είναι γενικού σκοπού – π.χ. δεν αναφέρεται σε κάποιο sprite
  - *“παίξε ένα τραγούδι όσο το παιχνίδι τρέχει"*
  - *“παρακολούθησε αν θα πατηθεί κάποιο κουμπί για να αλλάξει η κατάσταση του παιχνιδιού"*
  - *“σταμάτα το παιχνίδι μετά από Χ δευτερόλεπτα…"*

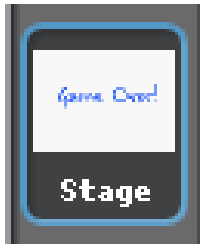
- Στο Scratch, το "stage" (σκηνή) μπορεί επίσης να περιέχει scripts
  - Αυτό είναι ένα πολύ καλό μέρος για να μπει ο κώδικας γενικού σκοπού…



Ένα μικρό demo…

lecture3_03.sb3

- Από την πρώτη άσκηση…
  - Ας κεντρικοποιήσουμε τον κώδικα και
  - ας βάλουμε ένα χρονόμετρο.



#3 Regression Testing

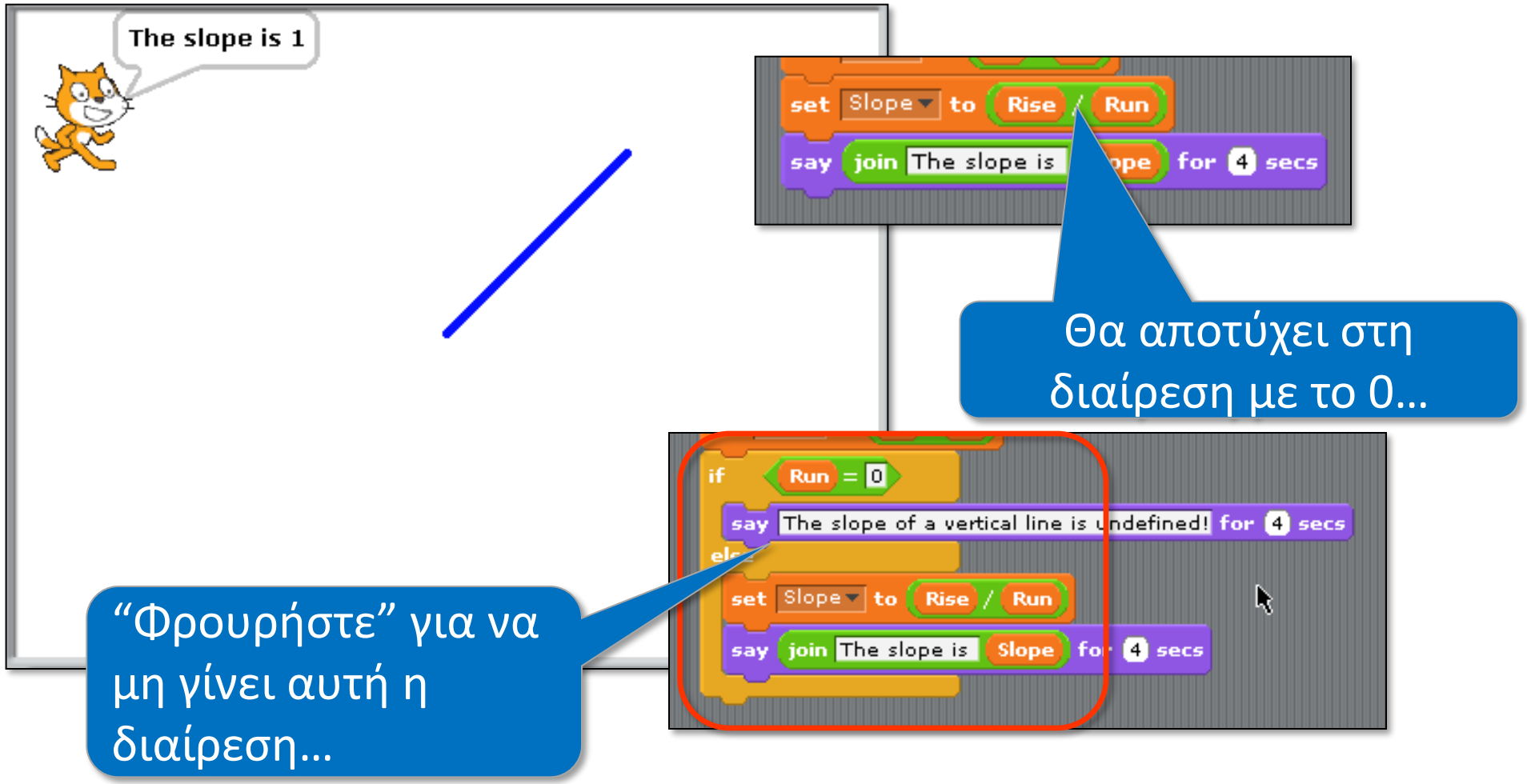
- Ο έλεγχος των προγραμμάτων είναι δύσκολος
  - Θα πρέπει να σκεφτείτε πολύ για όλα όσα το πρόγραμμά σας πρέπει να κάνει και να επινοήσετε ελέγχους, ώστε να σιγουρευτείτε ότι τα κάνει σωστά.
  - Θα πρέπει να γίνετε το ίδιο καλός/ή στο να «σπάτε» προγράμματα (να βρίσκετε τα λάθη τους), όσο είστε στο να τα δημιουργείτε.

- Έστω ότι επεξεργάζεστε και αλλάζετε το πρόγραμμά σας – τότε τι θα πρέπει να κάνετε;
  - Θα πρέπει να κάνετε **όλους τους ελέγχους ξανά**.
  - Αυτή η διαδικασία λέγεται **regression testing**.

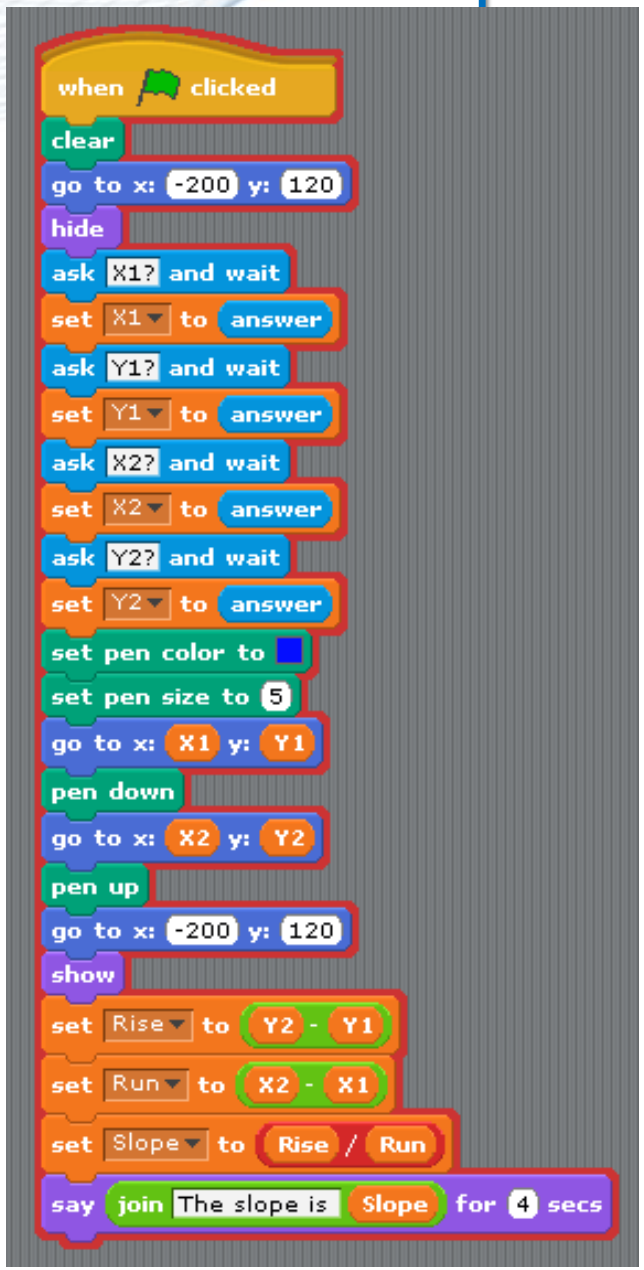
Κανόνας: θα πρέπει να διαθέσετε τόσο χρόνο στον έλεγχο του προγράμματος, όσο έχετε διαθέσει και στη δημιουργία του.

Ένα μικρό demo…

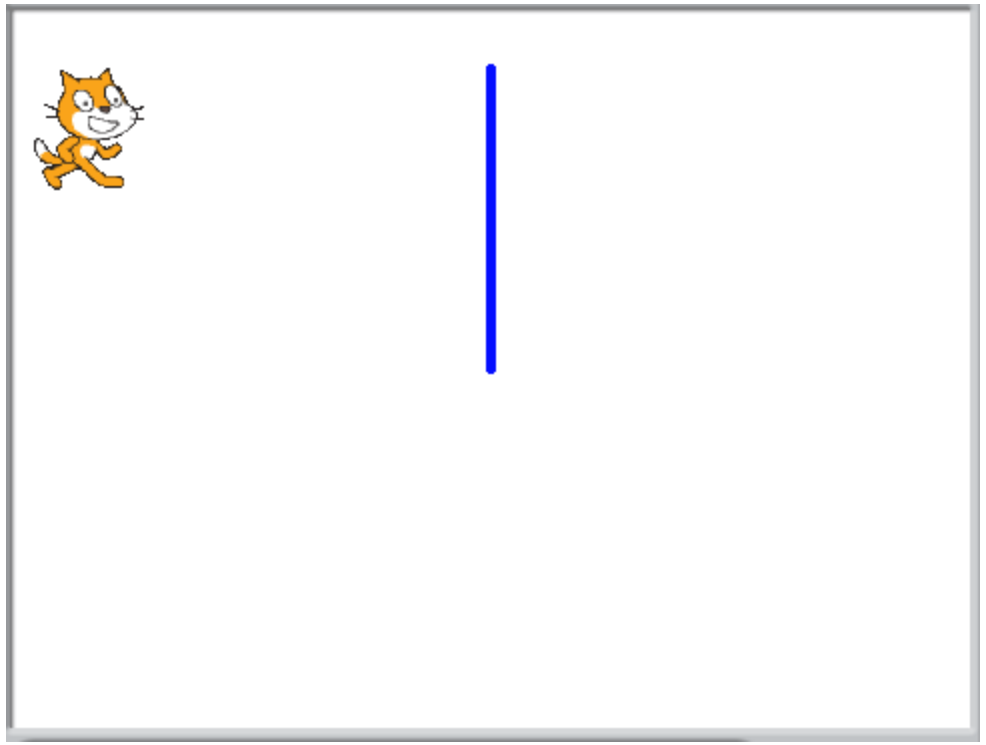
- Ας δούμε την άσκηση 2 με την κλήση….



Αν δεν κάνουμε τον έλεγχο…



lecture2_04.sb3



#4 Game Play

- Μία από τις προκλήσεις των παιχνιδιών είναι ο **συντονισμός** όλων των sprites.
  - *Όταν αλλάζουμε επίπεδα, πως το ξέρουν όλα τα sprites;*
  - *Όταν τελειώνει το παιχνίδι, πώς σταματάνε όλα τα sprites;*
  - *Πρέπει κάθε sprite να ξέρει πως τελειώνει το παιχνίδι, ή μπορούμε αυτό να το κεντρικοποιήσουμε;*
    - Ο κώδικας που υπάρχει κεντρικά μπορεί να αλλάξει ευκολότερα στο μέλλον…

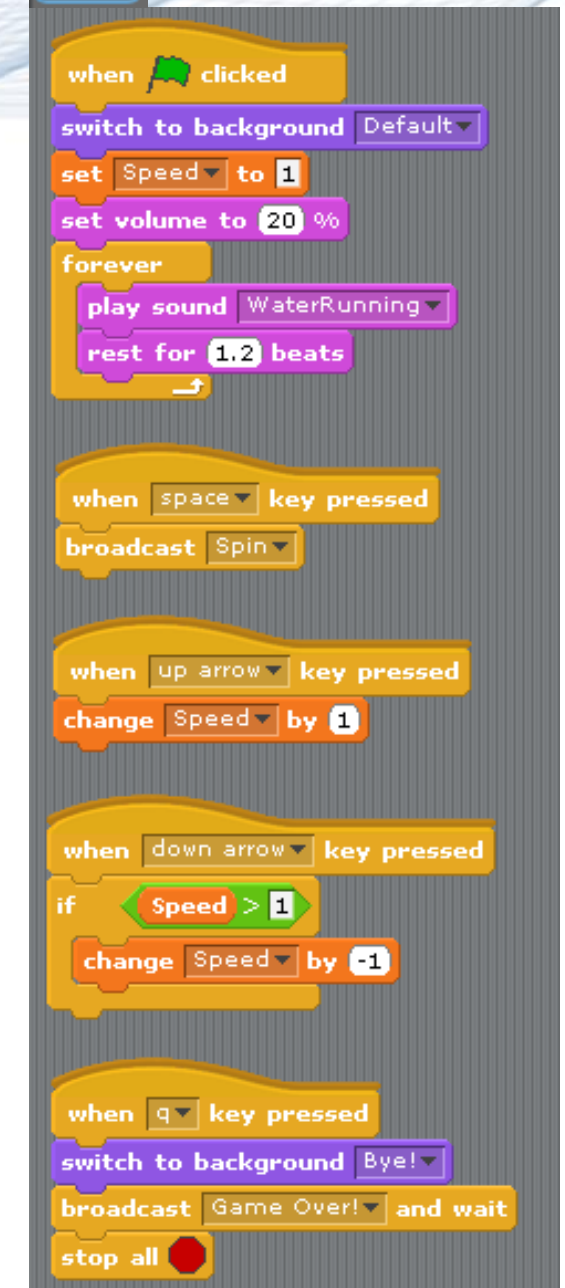
Λύση (Message Passing):

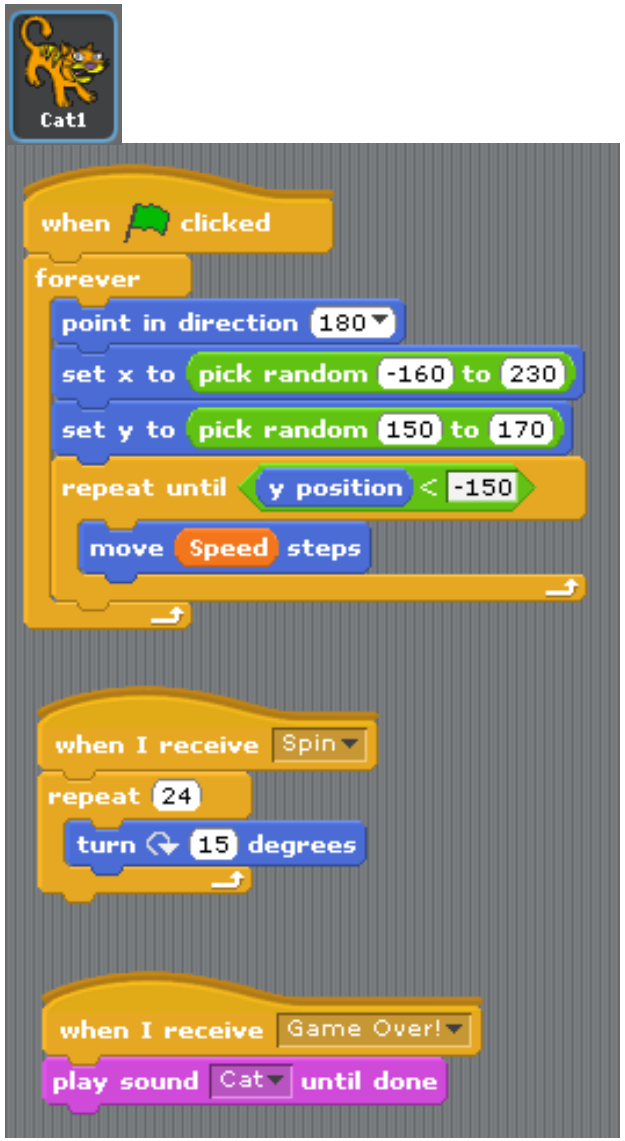
- χρησιμοποιείστε ένα **broadcast block** (στην κατηγορία *Events*) για να στείλετε ένα μήνυμα όταν κάτι συμβαίνει
- τα sprites χρησιμοποιούν ένα **when I receive block** (επίσης στην κατηγορία *Events*) για να αντιδράσουν σε ένα μήνυμα

Ένα μικρό demo…

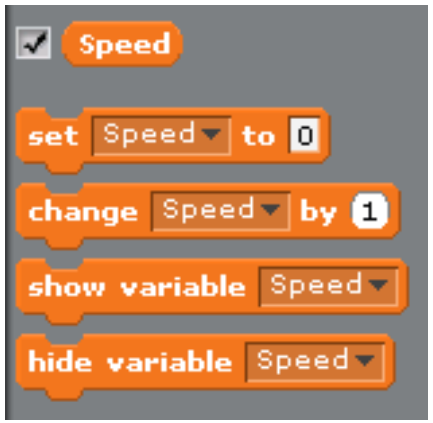
- Θα χρησιμοποιήσουμε το παράδειγμα με την καταιγίδα ("Raining Cats and Dogs") και θα:
  - Βάλουμε ένα "Game Over!" μήνυμα στο τέλος
  - Βάλουμε ένα "Spin" μήνυμα για να περιστρέφουμε τις γάτες και τους σκύλους
  - Βάλουμε κουμπιά αλλαγής ταχύτητας…

Στην πράξη…





lecture2_05.sb3



#5 Count, Sum, & Compute

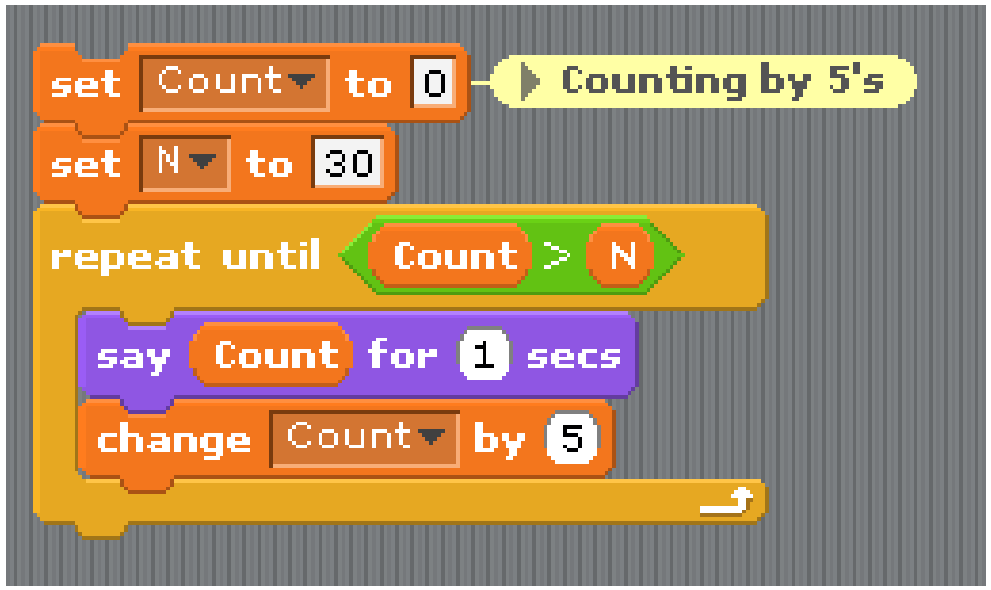
- Το Scratch είναι περισσότερο από μία γλώσσα δημιουργίας παιχνιδιών
  - Το Scratch είναι ικανό για γενικού σκοπού υπολογισμούς
  - Παράδειγμα: “Κλίση” το παράδειγμα που κάναμε

- Γενικά πρότυπα υπολογισμών:
  - μέτρηση-counting: 1, 2, 3, …, N
  - άθροιση-summing: 1 + 2 + 3 + … + N
  - υπολογισμοί-computing: πρόσθεση, πολλαπλασιασμός, τετραγωνική ρίζα, ημίτονο, συνημίτονο, …

Μέτρηση – Counting

- Υποθέστε ότι θέλετε να μετρήσετε από 0 .. N ανά 5
  - 0, 5, 10, 15, 20, 25, …

- ***Δημιούργησε*** *μία μεταβλητή "Count"*
- ***Δημιούργησε*** *μία μεταβλητή "N"*
- ***Θέσε*** *την Count σε 0*
- ***Θέσε*** *την N στην τελευταία τιμή*
- ***Repeat until*** *Count > N:*
  - *Κάνε τους υπολογισμού με την Count*
  - ***Άλλαξε*** *την Count κατά 5*

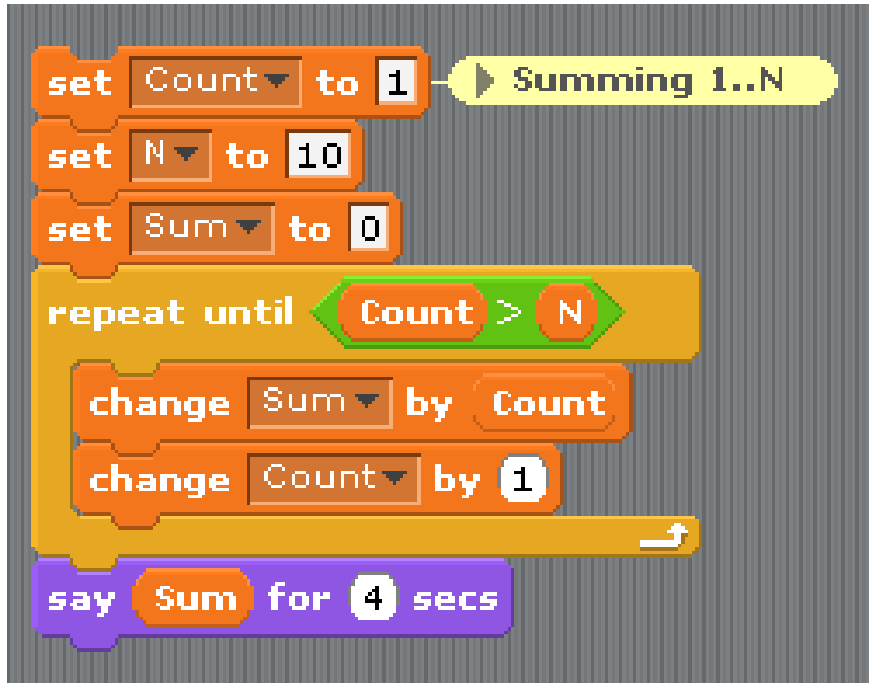


lecture2_06.sb3

Άθροιση- Summing

- Υποθέστε ότι θέλετε να αθροίστε τις τιμές 1 .. N
  - 1 + 2 + 3 + 4 + 5 + … + N

- ***Δημιούργησε*** *μία μεταβλητή "Count"*
- ***Δημιούργησε*** *μία μεταβλητή "N"*
- ***Δημιούργησε*** *μία μεταβλητή "Sum"*
- ***Θέσε*** *την Count σε 1*
- ***Θέσε*** *την N στην τελευταία τιμή*
- ***Θέσε*** *την Sum σε 0*
- ***Repeat until*** *Count > N:*
  - ***Άλλαξε*** *τη Sum κατά Count*
  - ***Άλλαξε*** *την Count κατά 1*



lecture2_06.sb3

Έλεγχος: sum(1..N) = $\frac{N*(N+1)}{2}$

Υπολογισμοί – Computing

- Παράδειγμα: Νόμοι της κίνησης του Νεύτωνα
  - Αυτή η εξίσωση υπολογίζει το ύψος Η ενός αντικειμένου, όταν το πετάμε ευθεία επάνω, χωρίς να υπολογίσουμε την αντίσταση του αέρα.

$$H = V_0 * T - 1/2g * T^2$$

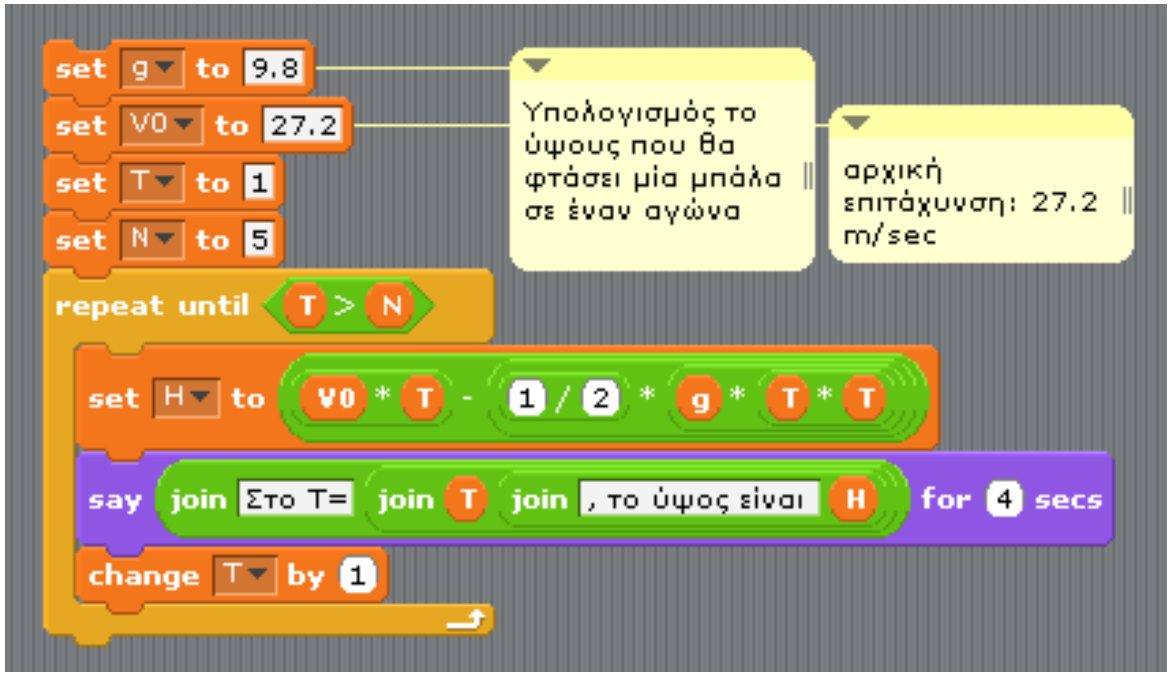
- ύψος
- Αρχική ταχύτητα
- χρόνος
- Επιτάχυνση της βαρύτητας ($9{,}8\ m/sec^2$)

Ας υπολογίσουμε το ύψος που θα φτάσει μία μπάλα σε ένα ποδοσφαιρικό αγώνα.

Νόμος της Κίνησης του Νεύτωνα

- Ας εφαρμόσουμε το νόμο:
  - Ο παίκτης πετάει τη μπάλα με περίπου 140m/sec
  - Ας υπολογίσουμε το ύψος με T=1, 2, 3, 4, 5

- ***Δημιούργησε*** *τις μεταβλητές που χρειαζόμαστε:*
  - *μέτρηση: T, N*
  - *υπολογισμός: H, V0, g*
- ***Θέσε*** *το g σε 9.8*
- ***Θέσε*** *το V0 σε 27.2*
- ***Θέσε*** *το T σε 1*
- ***Θέσε*** *το N σε 5*
- ***Repeat until*** *T > N:*
  - ***Θέσε*** *το H σε V0*T - ½*g*T*T*
  - ***Πες*** *H*
  - ***Άλλαξε*** *το T κατά 1*



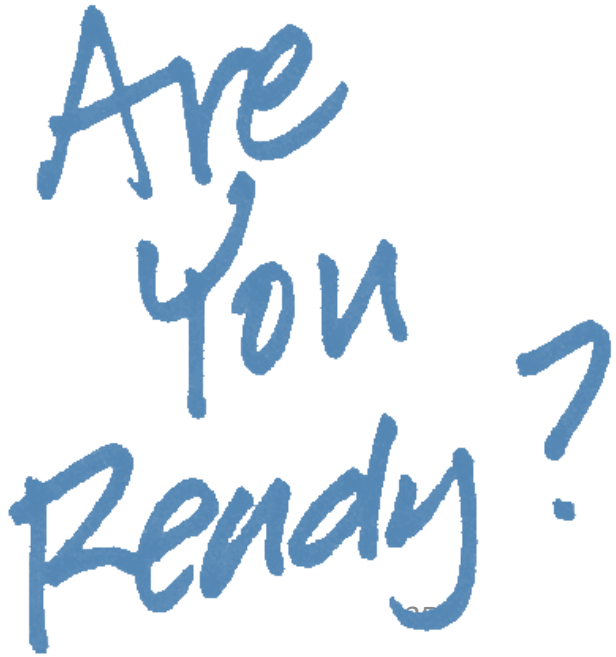
lecture2_06.sb3

Τι είδαμε μέχρι τώρα;

- Αυτό ήταν λοιπόν, είστε έτοιμοι! Έχετε αποκτήσει τις ακόλουθες δεξιότητες:
  - Αλληλουχία των εντολών
  - Επιλογή ανάμεσα σε διάφορες εναλλακτικές
  - Αξιοποίηση της Επανάληψης
  - Γνώση των μεταβλητών
  - Έχετε δει μερικά πρότυπα
  - Απλά χρειάζεστε περισσότερη εμπειρία και μεγάλη αποφασιστικότητα!!!

- Συνεχίστε να μαθαίνετε:
  - http://scratch.mit.edu/discuss/

- Για περισσότερη έμπνευση::
  - http://scratch.mit.edu/explore/projects/featured/



Και λίγη δουλειά για το σπίτι...

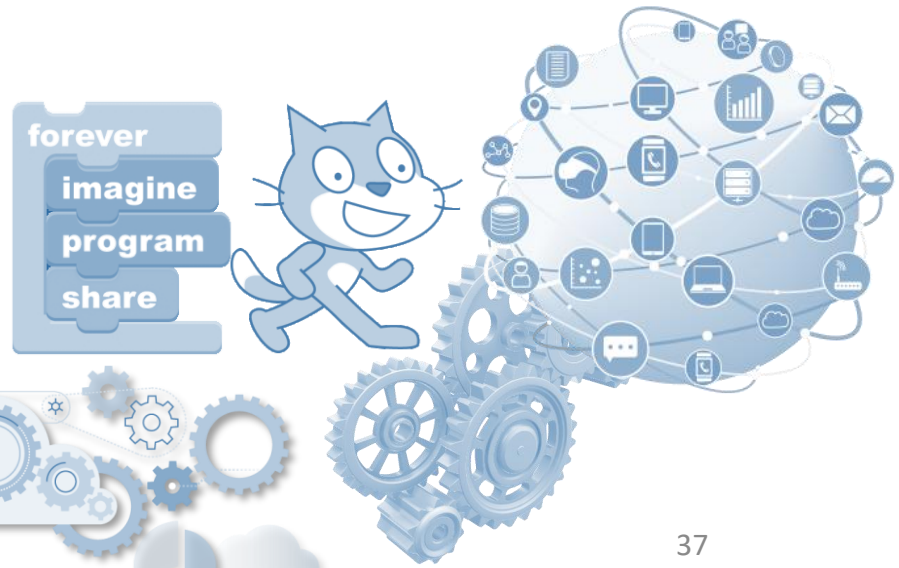
- **Υπολογίστε την πραγματική τροχιά μιας μπάλας.**

- Θεωρείστε ότι ο παίκτης στέκεται στο σημείο (0,0) και πετάει τη μπάλα με θετική x και y κατεύθυνση με μία γωνία A.

- Για να υπολογιστεί η θέση της μπάλας τη χρονική στιγμή T χρησιμοποιείστε τους ακόλουθους τύπους:
  - $x_T = \cos(A) * V0 * T$, $y_T = \sin(A) * V0 * T - 1/2 * g * T^2$

- Ο χρήστης θα πρέπει να δώσει τη γωνία σαν **είσοδο**.


how
which
why
what
when
where


forever
imagine
program
share